Homework Assignment 4
601.467 / 667 Introduction to Human Language Technology
Fall 2019
Due: Wednesday December 6th, 2019

November 22, 2019

Sequence Classification with Transfer Learning

In this assignment, you will implement two sequence classification models that predict if two sequence have a similar meaning or not.

Note: You may work in groups of (max) 3 people.

Question 1. Open the following colab notebook and complete the assignment:

https://colab.research.google.com/drive/1i9-uuizFuGJTzqgG$_0oP3eyiJNZ - M5gE$

Follow all the steps specified in it. Include link to your solved notebook in your submission. There are detailed `TODOs` specified in the colab link, you should complete all of them.

Report the following:

1. The number of params used by the LSTM based classifier:
2. Training and Validation loss for LSTM based classifier at the end of training. My implementation obtained a training and validation loss of 0.0614 and 1.0858.
3. Training and Validation accuracies for the LSTM based classifier at the end of training. My implementation gets training and validation accuracy of 97.79% and 75.62%.
4. The number of params used by the BERT based model:
5. Training and Validation loss for the BERT based model. My implementation attains $0.2145, 0.3268$ for training and validation loss at the end of 3 epochs.
6. Training and Validation accuracies for the BERT based model. My model attains $91.51\%, 87.57\%$ training and validation accuracy at the end of 3 epochs.

7. Test Prediction Accuracy for the LSTM based model:

8. Test Prediction Accuracy for the BERT based model:

Answer the following(In 3-5 sentences):

1. We have talked about word2vec and BERT in class, how are these different?

2. How would you incorporate pretrained word2vec representation into the LSTM based model that you have built above?

3. Could you incorporate pretrained word embeddings into the BERT based model? Do you think it would help? Justify your answer.

4. Suppose you were not allowed to fine-tune the pretrained BERT model, could you still find a way to build a sentence similarity classifier? Describe your approach. Hint: look up special tokens in BERT (read Tokenizer Docs and the example inputs from here)

Question 2. (Optional) Can you modify your LSTM based Classifier to use two bidirectional LSTMs instead of unidirectional? You would also need to modify the linear layer to make the bidirectional version work. Does this make improve the performance of your classifier?